

# DESENVOLVIMENTO DE UM COMPILADOR PARA A LINGUAGEM LUNA: IMPLEMENTAÇÃO DE ANÁLISE LÉXICA, SINTÁTICA, SEMÂNTICA E INTERPRETAÇÃO

JOÃO LUCAS DE OLIVEIRA SILVA

Unic Beira Rio

Palavras-chave: Compiladores; Análise Léxica; Análise Sintática; Análise Semântica; Interpretação de Código.

## 1. Introdução

O desenvolvimento de compiladores é uma prática essencial para compreender os mecanismos internos das linguagens de programação. Este projeto apresenta a construção de um compilador didático em Python para uma linguagem fictícia chamada LUNA, abordando todas as etapas fundamentais: análise léxica, análise sintática, representação intermediária (AST), análise semântica e interpretação.

A linguagem LUNA foi projetada com uma sintaxe simples e clara, permitindo a implementação de estruturas básicas como declarações de variáveis, expressões aritméticas e lógicas, condicionais `if-else` e comandos de saída `print`.

## 2. Objetivos

### 2.1. Objetivo Geral

Desenvolver um compilador funcional em Python para a linguagem LUNA, aplicando os

conceitos teóricos de compilação em um sistema prático e modular.

### 2.2 Objetivos Específicos

- a. Implementar um analisador léxico baseado em expressões regulares.
- b. Construir um parser recursivo descendente com suporte à precedência e associatividade.
- c. Desenvolver uma Árvore Sintática Abstrata (AST) para representar o código-fonte.
- d. Implementar uma Tabela de Símbolos para análise semântica.
- e. Verificar tipos e escopo durante a análise semântica.
- f. Criar um interpretador que execute diretamente a AST.
- g. Validar o compilador com testes automatizados.
- h. Documentar todas as etapas do desenvolvimento.

# DESENVOLVIMENTO DE UM COMPILADOR PARA A LINGUAGEM LUNA: IMPLEMENTAÇÃO DE ANÁLISE LÉXICA, SINTÁTICA, SEMÂNTICA E INTERPRETAÇÃO

JOÃO LUCAS DE OLIVEIRA SILVA

Unic Beira Rio

Palavras-chave: Compiladores; Análise Léxica; Análise Sintática; Análise Semântica; Interpretação de Código.

## 3. Metodologia

A metodologia adotada foi a de pesquisa experimental aplicada, com foco na construção incremental de um compilador em Python. O projeto foi dividido em seis fases principais:

### 3.1 Análise Léxica

Utilizou-se expressões regulares para identificar tokens como números, identificadores, operadores, parênteses, chaves, palavras-chave (if, else, print) e símbolos de pontuação. O analisador léxico transforma o código-fonte em uma lista de tokens.

### 3.2 Análise Sintática

Foi implementado um parser recursivo descendente que constrói a AST a partir dos tokens. A gramática da linguagem ELE foi definida com suporte à precedência de operadores e agrupamento por parênteses.

### 3.3 Representação Intermediária (AST)

A AST foi modelada com classes Python que representam os diferentes elementos da linguagem: números, variáveis, operações binárias, atribuições, comandos print, estruturas if-else e blocos de código.

### 3.4 Análise Semântica

A análise semântica verifica a existência e o tipo das variáveis por meio de uma Tabela de Símbolos. Erros como variáveis não declaradas ou operações inválidas são detectados nessa etapa.

### 3.5 Interpretação

O interpretador percorre a AST e executa os comandos diretamente. Expressões são avaliadas, variáveis são armazenadas e comandos print exibem os resultados no console.

### 3.6 Validação e Testes

Foram desenvolvidos testes automatizados para validar cada etapa do compilador. Os testes incluem casos positivos e negativos,

# DESENVOLVIMENTO DE UM COMPILADOR PARA A LINGUAGEM LUNA: IMPLEMENTAÇÃO DE ANÁLISE LÉXICA, SINTÁTICA, SEMÂNTICA E INTERPRETAÇÃO

JOÃO LUCAS DE OLIVEIRA SILVA

Unic Beira Rio

Palavras-chave: Compiladores; Análise Léxica; Análise Sintática; Análise Semântica; Interpretação de Código.

como uso correto de variáveis e detecção de erros semânticos.

- Arquitetura modular e expansível

## 4. Resultado

O compilador LUNA em Python foi concluído com sucesso, implementando todas as funcionalidades previstas. A linguagem suporta:

- Declarações e atribuições de variáveis
- Expressões aritméticas e lógicas
- Estruturas condicionais `if-else`
- Comandos de saída `print`

Durante os testes, o sistema demonstrou:

- Correção sintática e semântica
- Feedback imediato na execução

Foram realizados 5 testes automatizados, todos aprovados, incluindo casos de erro como uso de variáveis não declaradas e divisão por zero.

## 5. Conclusão

O desenvolvimento do compilador LUNA em Python proporcionou uma experiência prática valiosa, consolidando os conceitos teóricos de compilação. A separação clara entre as etapas do compilador facilitou a manutenção e permitiu futuras extensões, como suporte a laços de repetição, funções e arrays.

Este projeto reforça o valor pedagógico da construção de compiladores como ferramenta de aprendizado em cursos de ciência da computação e engenharia de software.

# DESENVOLVIMENTO DE UM COMPILADOR PARA A LINGUAGEM LUNA: IMPLEMENTAÇÃO DE ANÁLISE LÉXICA, SINTÁTICA, SEMÂNTICA E INTERPRETAÇÃO

JOÃO LUCAS DE OLIVEIRA SILVA

Unic Beira Rio

Palavras-chave: Compiladores; Análise Léxica; Análise Sintática; Análise Semântica; Interpretação de Código.

## Referências

- Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). *Compilers: Principles, Techniques, and Tools*. Pearson.
- Carmona, J. V. (2011). *Análise e Desempenho de Tráfego Triple Play em Redes PLC de Baixa Tensão*.